Using an IP Container Demo Script

Introduction

This demonstration introduces the IP flow, including the core container feature.

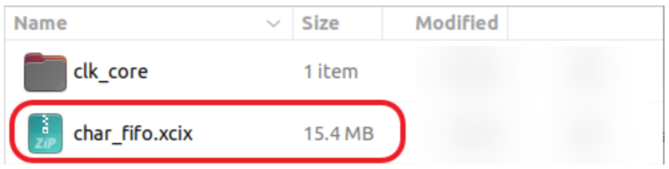
Preparation:

- Required files: `$TRAINING_PATH/Core_Container/demo/KCU105/verilog`
- Required hardware: None
- Supporting materials: None

Using an IP Container

Action with Description	Point of Emphasis and Key Takeaway
• Launch the Vivado™ Design Suite. • Unzip the project using the Tcl Console: `exec unzip $::env(TRAINING_PATH)/Core_Container/demo/KCU105/verilog.zip -d $::env(TRAINING_PATH)/Core_Container/demo/KCU105/verilog`	
• Open the **wave_gen.xpr** project from the following directory: `$TRAINING_PATH/Core_Container/demo/KCU105/verilog`	• You can easily open an existing Vivado IDE project via the Getting Started page.

Action with Description	Point of Emphasis and Key Takeaway
• Go to **IP Sources** in the Sources window. • Examine the output products that are generated. • Expand the **char_fifo** and **clk_core** folders if needed.	• The IP Sources window shows the generated output products. • By default, our IP will generate: • An instantiation template • Synthesis files ▪ Constraints files associated with synthesis • Simulation files ▪ Simulation sources associated with IP • Change log • Design checkpoint (.dcp) ▪ Synthesizes the IP and generates the DCP file in OOC mode • Simulation netlist and stub files
• Observe the directory structure of the generated output products for the IP. • Browse to the `$TRAINING_PATH/Core_Container/demo/KCU105/verilog/wave_gen.srcs/sources_1/ip` directory. • Notice that there are two directories for each IP.	• The *clk_core* directory contains the output products for the *clk_core* IP. • The *char_fifo* directory contains the output products for the *char_fifo* IP.
• Enable the core container feature for the *char_fifo* IP and observe the single file (*.xcix*) representation for the IP. • Right-click **char_fifo** in the IP Sources window and select **Enable Core Container**. • Browse to the `$TRAINING_PATH/Core_Container/demo/KCU105/verilog/wave_gen.srcs/sources_1/ip` directory. • Notice the XCIX file that is generated.	• The core container feature helps simplify working with revision control systems by providing a single file representation of an IP. • This optional feature allows you to have IP and all generated output files contained in one compressed binary file with the extension XCIX. • When the core container feature is enabled for an existing IP, the XCIX file replaces the IP directory and the output products. • Enabling the core container for an IP changes the on-disk representation of that IP instance; the internal representation for the IP remains the same within the Vivado Design Suite.

Action with Description	Point of Emphasis and Key Takeaway
 Name / Size / Modified; clk_core — 1 item; char_fifo.xcix — 15.4 MB	
• Examine the output products for the *char_fifo* IP in the IP Sources tab of the Sources view. • Select the **IP Sources** tab in the Sources window. • Expand the **char_fifo** and **clk_core** folders and observe that the output products for both IPs are the same.	• Within the Vivado IDE Sources view the two IP appear the same, both listing the output products, all of which can be opened for viewing from within the Vivado IDE. • On disk there is a folder for *clk_core* and the single XCIX file for the *char_fifo* IP. • When an IP uses core container, the Vivado Design Suite reads the IP source files needed during synthesis and implementation from this single XCIX file. The files are not extracted to a temporary directory; they are read directly from the XCIX file.
• Select **File** > **Exit** to close the Vivado Design Suite.	

Summary

This demonstration showed you how to use the core container feature.

References:

- Supporting materials
  - *Vivado Design Suite User Guide: Designing with IP* (UG896)
  - Using Core Containers for IP Quick Take video (https://www.xilinx.com/video/hardware/using-core-containers-for-ip.html)